Rotations from Shears: A Useful Computational Viewpoint

Introduction

A rotation is usually presented as a single matrix operation. In two dimensions this is a 2×2 matrix; in three dimensions it is a 3×3 matrix. The standard viewpoint is that a rotation is an indivisible geometric primitive.

An alternative viewpoint is that a rotation can be built from a sequence of much simpler operations called *shears*. A shear leaves one coordinate unchanged while shifting another coordinate by an amount proportional to it. For example,

$x \leftarrow x + ky$

is a shear. Geometrically, squares become parallelograms, but areas are preserved.

This decomposition is more than a mathematical curiosity. It often leads to algorithms that are simpler, more efficient, more local in memory access, and sometimes numerically more robust than direct matrix multiplication.

The idea is well known in two dimensions, where a rotation can be decomposed into three shears. A 1997 paper by Tommaso Toffoli and Jason Quick showed that a general three-dimensional rotation can also be decomposed into only three carefully chosen shears.

Why Shears Are Interesting

A shear has several useful properties:

- Determinant exactly equal to 1.
- Volume and area are preserved exactly.
- The inverse is another shear.
- Implementation requires only additions and multiplications.
- Data movement is local and structured.

Unlike a general matrix transformation, a shear updates only one coordinate at a time. This makes shears attractive for image processing, simulation, hardware implementation, and numerical algorithms.

For example,

$x \leftarrow x + ay + bz$

is a 3D shear (x is updated, y and z are not.) More generally, a shear updates one coordinate using one or more of the others while leaving the remaining coordinates unchanged.

Three-Dimensional Rotations from Three Shears

The main result of the Toffoli–Quick paper is that an arbitrary 3D rotation can be written as

Rotation = Shear3 × Shear2 × Shear1

rather than as a sequence of many smaller rotations.

The construction begins with a standard Euler-angle representation of the rotation and then algebraically rearranges the factors until only three shear matrices remain.

For implementation purposes the important fact is not the exact formulas but the structure:

1. Apply a shear along one coordinate direction.
2. Apply a second shear along another direction.
3. Apply a final shear.

The combined effect is exactly the desired rotation.

Thus a geometrically complex operation is replaced by three simple coordinate updates.

Why This Matters

The original motivation came from image processing.

A direct image rotation requires interpolation because rotated pixels do not generally land on integer pixel coordinates. Shears are easier to implement because rows, columns, or slices can be shifted independently.

In three dimensions the same idea applies to voxel volumes. Rather than performing a complicated global transformation, the volume is processed as a sequence of simple directional shifts.

The resulting algorithms are naturally suited to hardware, memory systems, and parallel processing architectures.

A Broader Result

The deeper contribution of the paper is not actually about rotations.

The authors show that almost any volume-preserving linear transformation can be decomposed into three shears.

Symbolically,

A = U L V

where

- U is an upper-triangular shear,
- L is a lower-triangular shear,
- V is another upper-triangular shear.

Thus rotations are merely one member of a much larger family.

This suggests a useful algorithmic strategy:

Instead of asking how to implement a complicated transformation directly, ask whether it can be factored into a small number of shears.

Often the answer is yes.

Connections to Other Numerical Methods

This viewpoint appears repeatedly across computational mathematics.

LU Factorization

A matrix can be decomposed into triangular factors that are easier to apply and invert.

Symplectic Integrators

Many physics simulators split a difficult update into simple volume-preserving substeps. These preserve energy and stability much better than naive methods.

The Vicanek Oscillator

The Vicanek quadrature oscillator can be interpreted as a sequence of shears rather than a direct rotation. This explains several of its desirable properties:

- exact area preservation,
- simple inverse,
- excellent numerical stability,
- low computational cost.

Invertible Neural Networks

Many modern invertible architectures use coupling layers that are essentially nonlinear shears. These preserve invertibility while remaining easy to compute.

Beyond Linear Shears

The idea extends naturally beyond matrices.

Instead of

$x \leftarrow x + ay$

one may use

$x \leftarrow x + f(y)$

or

$x \leftarrow x + f(y,z)$

for some nonlinear function f.

These nonlinear shears often remain exactly invertible because only one coordinate is modified at a time.

Such constructions appear in:

- invertible neural networks,
- volume-preserving dynamical systems,
- chaos and mixing algorithms,
- reversible computation.

In this broader setting, shears become a fundamental building block for constructing complex transformations from simple invertible pieces.

Conclusion

The traditional view treats rotation as a primitive operation.

The shear viewpoint treats rotation as a composition of simpler operations that preserve important invariants. The 3D rotation decomposition of Toffoli and Quick demonstrates that this approach remains practical even in higher dimensions.

More importantly, it points toward a general design principle:

When confronted with a complicated transformation, search for a decomposition into a small number of invertible shears.

The resulting algorithm is often simpler, more robust, easier to invert, and more amenable to efficient implementation than the original formulation.

Appendix:

3D rotation:

Convert your rotation to Euler angles

Use

$R = R_3(a) R_2(b) R_3(c)$

where:

- a = first Z rotation
- b = Y rotation
- c = second Z rotation

Build three shear matrices

First shear:

$$S_1 = \begin{bmatrix} 1 & -\tan((a+c)/2) & \cos((a-c)/2)\tan(b/2) \\ 0 & 1 & -\sin(c)\tan(b/2) \\ 0 & 0 & 1 \end{bmatrix}$$

Second shear:

$$S_2 = \begin{bmatrix} 1 & 0 & 0 \\ \sin(a+c) & 1 & 0 \\ -\frac{\cos(c)\sin(b)}{\cos((a+c)/2)} & -\frac{\sin((a-c)/2)\sin(b)}{\cos((a+c)/2)} & 1 \end{bmatrix}$$

Third shear:

$$S_3 = \begin{bmatrix} 1 & -\tan((a+c)/2) & \cos(a)\tan(b/2) \\ 0 & 1 & \sin(a)\tan(b/2) \\ 0 & 0 & 1 \end{bmatrix}$$

Apply the shears

For a point:

p1 = S1 * p

p2 = S2 * p1

p3 = S3 * p2

The result equals the full rotation.